A Two-Step Approach for Explainable Relation Extraction

H. Ambre Ayats, Peggy Cellier, Sébastien Ferré

A Two-Step Approach for Explainable Relation Extraction

Hugo Ayats, Peggy Cellier, and Sébastien Ferré

Univ Rennes, INSA, CNRS, IRISA
Campus de Beaulieu, 35042 Rennes, France
{hugo.ayats,peggy.cellier,sebastien.ferre}@irisa.fr

Abstract. Knowledge Graphs (KG) offer easy-to-process information. An important issue to build a KG from texts is the Relation Extraction (RE) task that identifies and labels relationships between entity mentions. In this paper, to address the RE problem, we propose to combine a deep learning approach for relation detection, and a symbolic method for relation classification. It allows to have at the same time the performance of deep learning methods and the interpretability of symbolic methods. This method has been evaluated and compared with state-of-the-art methods on TACRED, a relation extraction benchmark, and has shown interesting quantitative and qualitative results.

1 Introduction

Knowledge Graphs (KG) [10] have the advantage to offer easy-to-process information. However, most available information is still in the form of texts. A key problem is therefore the extraction of KGs from text, which amounts to identify named entities and relationships [16]. Relation Extraction (RE) [9] is the sub-problem of identifying and labelling relationships, assuming that the named entities have already been identified. Currently, the best scores on RE are achieved by deep learning methods, such as LUKE [22] or BERT [4]. While their scores have recently increased significantly (e.g., F1-score 72.7 for LUKE), the KGs that would result from their systematic application would still be noisy and incomplete to a large extent (e.g., 30% incorrect triples, and 25% missing triples for LUKE). Therefore, a completely automated process does not seem realistic if we aim at reliable and complete KGs and the RE task is too tedious to perform by hand only.

It seems necessary to introduce some human control in the extraction process while providing support for automation. Our idea is to base the automation on an increasing set of extraction rules, which are generated from previous examples and validated by humans. Human validation ensures the reliability of the extracted KG, and the generic aspect of rules supports the automation of the information extraction process. In this paper, we focus on the sub-task of generating extraction rules from *examples*, i.e. sentences in which relationships have already been identified and labelled. Unfortunately, deep learning methods only predict relationships at the instance level, they do not provide information that can be leveraged into general and interpretable extraction rules. In previous work [1] a symbolic approach based on Concepts of Neighbours [5] was proposed to provide explainable predictions. Those explanations have the potential to be translated into extraction rules. However, it only solves the sub-problem of *relation classification*, i.e. when the relationships have already been detected and only remains to be labelled. Indeed, explanations can be found for the label of a relationship but hardly for the absence of a relationship as there are many ways for two entities not to be in relationship.

To address the RE problem, we propose to combine a deep learning approach for relation detection, and the symbolic approach based on Concepts of Neighbours for relation classification. It allows to have at the same time the performance of deep learning methods and the interpretability of symbolic methods. We conducted experiments showing that in terms of F1-score on the full RE task, our composite approach is comparable to deep learning approaches using the same kind of information from texts (i.e., syntactic structure, lexical semantics), namely GCN and C-GCN [23]. In contrast to deep learning methods, our approach generates explanations for each prediction, and convert them into extraction rules. Those extraction rules exhibit rich structures, mixing different levels of information from texts: lexical, syntactical, and semantic. In addition, they are generalizations of the current prediction, which makes them useful for the automation of future extractions.

2 Related Works

Most approaches addressing the Relation Extraction task use deep learning methods. Historically, convolutional neural networks [19] and LSTM [21] were used first, then were replaced by graph convolution networks methods [23], which allow to take into account the syntactic structure of sentences. Currently, the approaches that give the best results for the RE task use pre-trained language models such as BERT [4] and its variants [11,22]. However, the performance of those approaches (with an F-score between 70 and 75% on the TACRED benchmark [24]) are still too low to allow a full automation. In addition, those fully statistical approaches lack of explanations for their predictions, which limits the possibilities of introducing human control in the process to improve reliability.

Symbolic approaches have also been proposed for the RE task. Their performance are often lower than deep learning methods, but by definition they provide interpretable results that can be used in a process with human control. The first symbolic approaches use rules such as regular expressions [8] or syntactic patterns [7]. However, these rules are handcrafted and thus those approaches are time consuming and often devoted to a specific corpus. Some symbolic approaches automatically learn the linguistic rules. For instance [3] uses pattern mining techniques to automatically extract those rules. The method presented in [2] combines symbolic and machine learning techniques and proposes to learn patterns from a list of seed terms, i.e. pairs of entities known to be in some tar-

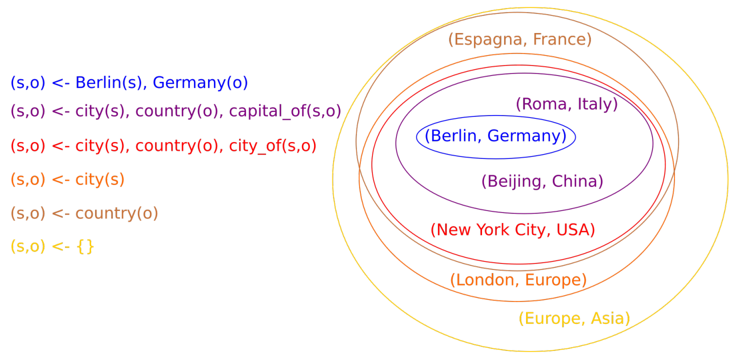

Fig. 1. Example of Concepts of Neighbours

get relation. More recently, two symbolic approaches based on Formal Concept Analysis (FCA) have been proposed to populate a KG from texts [12,1]. The latter is based on Concepts of Neighbours, which have also been used for KG completion [6].

3 Relation Classification with Concepts of Neighbours

In this section, we describe the use of Concepts of Neighbours for the problem of explainable relation classification. Given a sentence (e.g., "Berlin became the capital of Germany in 1990"), two named entities in the sentence (e.g., "Berlin" and "Germany"), and the assumption that there is a relationship between the two entities, the problem is to predict the label of the relationship (e.g., "is the capital of"), and to provide interpretable explanations for the predicted label. The work presented in this section is developed in more details in [1].

3.1 Concepts of Neighbours

Concepts of Neighbours [5] is a graph mining method for entity-relation graphs that aims, for a given tuple of entities, to compute which are the most similar tuples of entities. It can be seen as a symbolic form of the k-nearest neighbours method, where numeric distances are replaced by common graph patterns. The bigger the common graph pattern between two tuples, the closer they are. For example, suppose that we want to find couple of entities similar to *(Berlin, Germany)* in a graph about geography. Concepts of Neighbours hierarchically clusters all couples of entities into *concepts* according to their similarity with *(Berlin, Germany)*. Figure 1 shows the set of concepts as a Venn diagram. Each concept is defined by its *intension*, which is a graph pattern with distinguished variables, and its *extension*, which is the set of couples matching the intension. It can be seen that *(Roma,Italy)* is a close neighbour as it shares the "capital of" relation, while *(New York,USA)* is a farther neighbour because New York is

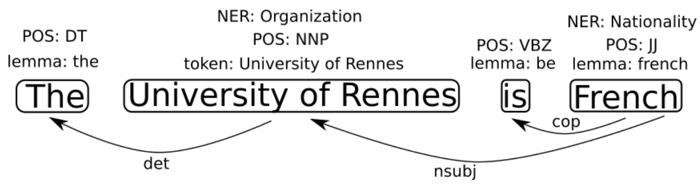

Fig. 2. Example of sentence modeling

only a city of USA. The *proper extension* of a concept is defined as the subset of tuples of its extension that are not in the extension of more specific concepts. The *extensional distance* is defined as the size of its extension, and can be used as a numerical distance.

3.2 Application to Text

In order to apply Concepts of Neighbours to texts, we first need to model a text as an entity-relation graph. Figure 2 shows the modeling of the sentence "The University of Rennes is French". We rely on NLP tools and resources to extract syntactic and semantic information from text[1].

The graph representing each sentence is defined as follows. *Tokens* are used as entities, and are linked by the *dependency relations*. *Lemmas*, *named entity types* and *part-of-speech (POS) tags* are then added as entity labels.

From there we apply a few enhancements to the graph. First, some named entities extend over several tokens but have a syntactic and semantic unity: e.g. "University of Rennes" is split in three tokens. We decided to merge those tokens into a single entity in our graph representation, and to label it with the concatenation of tokens instead of using the lemmas, considering them as proper nouns. Second, we enrich the graph labelling following syntactic and semantic inferences. The objective is to help finding common graph patterns with Concepts of Neighbours. For instance, on the syntactic side, singular nouns have POS tag NN whereas plural nouns have POS tag NNS. To relax the singular/plural distinction, we infer POS tag NN for every entity that has POS tag NNS. On the semantic side, given an entity labelled with some lemma (e.g. "school"), we infer labels for the synonyms and hypernyms of the lemma (e.g., "educational institution")[2]. The Concepts of Neighbours method is capable of handling such inferences efficiently, without having to materialize them in the graph, by relying on a partial ordering over the entity and relation labels.

3.3 Application to Relation Classification

Given the graph modeling of a text, and the choice of a couple of named entities $(subject, object)$, Concepts of Neighbours can compute a set a concepts of neighbours, each concept being associated with a set of neighbour couples (the proper

[1] We decided to use the *Stanford CoreNLP* toolkit [15]
[2] We used *WordNet* [18] to do so

extension), and to an extensional distance. From there, a label of the relation from *subject* to *object* can be predicted by looking at the relationships holding for the neighbour couples of each concept c. Intuitively, the more neighbours in the proper extension of c hold some relation r, and the smaller the extensional distance of c, then the stronger the prediction for relation r is. This is formalized as the confidence of the rule $R_{r,c} : P_c \rightarrow r(s,o)$, where $(s,o) \leftarrow P_c$ is the intension of concept c.

$$conf(R_{r,c}) = \frac{|\{(s,o) \mid r(s,o)\} \cap ext(c)|}{ext\_dist(c)}$$

To aggregate the rules from all concepts to all relations and to get a ranking of predicted relations, we use *Maximum Confidence* [17], which was applied with success for link prediction [17,6]. Informally, the predicted relation is the relation which has the higher maximal confidence. In case of equality, the predicted relation is the one with the higher second maximal confidence, and so on.

In practice, the generic prediction method presented above is specialized to the settings of relation extraction benchmarks like TACRED. First, neighbours are only searched among the couples of entities that are annotated by a relation that is compatible with the entity types, according to the RECENT paradigm [13]. Second, we apply the pruning strategy proposed in [23], where only tokens that are at a maximal distance k of the path between the subject and object are kept in the representation of a sentence.

4 A Two-Step Approach for Relation Extraction

The method presented in the previous section works by similarity, classifying test examples among the different relations according to similar training examples. If it works for deciding *which* relation exists between a subject and an object, it does not work for knowing *if* a relation exists. Indeed, there is no reason for a negative example (*i.e.* an example with no relation) to look like other negative examples. Therefore, this method can perform *relation classification* but cannot perform *relation detection*. However, those two steps are necessary to perform proper relation extraction.

The idea is to combine two methods, one for relation detection only, and the method presented in the above section for relation classification. Figure 3 presents how such a system works: the method for relation detection discriminates the positive examples from the negative examples, and our neighbours-based method classifies the positive examples among the compatible relation types. Such a two-step approach has already been exploited with promising results [14].

4.1 Relation Detection with Deep Learning

As there is no efficient symbolic or fully explainable method for relation detection that we know of, we decided to favor performance and therefore to use a state-of-the-art deep learning approach. Moreover, there is not much need to explain the

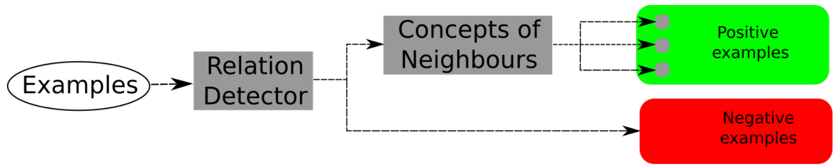

Fig. 3. Two-step relation classification process

non-existence of a relation, and an explanation for a type of relation is also an explanation for the existence of a relation. Today, the state-of-the-art in relation extraction is dominated by pre-trained language models such as BERT [4] and its variants. One of those variants, LUKE [22], has the particularity to handle both single words and multi-word entities, and has shown impressive results on diverse NLP tasks, including relation extraction. We decided to use this model as a relation detector.

We consider several configurations of LUKE for relation detection. The first one, called *luke-base*, simply consists in reusing the fully trained model for relation extraction and post-process the output in order to merge all the positive predictions into one class. A second configuration, called *luke-detect* consists in specializing LUKE for relation detection. We remove LUKE's last classification layer, and replace it by two layers: a fully connected layer of size n and an output neuron with a *sigmoid* activation function. Then the model is fine-tuned in order for it to predict 1 on the positive examples and 0 on the negative examples.

4.2 Explainability

The main asset of this two-step method is its explainability: for a given prediction, if this prediction is not *no_relation*, the method is able to provide an explanation. This limitation to positive prediction may seem odd, but this can be understood by the fact that if it is easy to imagine how to explain why there is a relation between two examples (by giving other annotated examples looking like the given example), it is more complicated to explain why a given example has no relation between its subject and its object, as negative examples do not have to look like other negative examples.

For a given example annotated as positive, the raw explanation that can be given is the whole set of Concepts of Neighbours of this example. However, whereas it is a complete explanation, it is hardly readable for a non-expert. However, among this set of concepts, only a few ones are used to make a prediction: the ones that have an intension which was used to create a rule of maximum confidence. Therefore, by displaying those intensions and the examples matching it, we obtain a short and readable explanation (only a few graph patterns and the related sentences).

Table 1. Precision, recall and F-score for relation detection methods

Approach	P	R	F1
luke-base	74.8	79.9	**77.3**
luke-reprod	**76.8**	75.2	76.0
luke-redetect	73.1	**80.1**	76.4

5 Experiments and Results

In this section, we present the different experiments made with our relation extraction system and the subsequent results. Those experiments can be divided in three parts: 1) the LUKE-based Relation Detection module, 2) the Concepts of Neighbours-based Relation Classification module, and 3) the whole system.

All experiments were made on the TACRED dataset [24], one of the most used dataset for Relation Extraction. This dataset is made of 106,264 examples, split into a training corpus (68,124 examples), a development corpus (22,631 examples) and a test corpus (15,509 examples). Each example of this dataset is a sentence with two entity mentions (a subject and an object), each mention being typed among 23 possible types, and annotated with a relation type among 41 effective classes plus a *no_relation* class representing the absence of relation between the subject and the object. For greater accuracy compared to random pairs of entity mentions occurring in real-world sentences, 79.5% of the examples are in the *no_relation* class.

5.1 Relation Detection

We evaluate the different configurations of LUKE [22] presented in Section 4.1, in order to choose the best one for relation detection.

Experiment Design As presented in Section 4.1, several configuration of LUKE were tested. In addition to *luke-base* and *luke-detect* , a third configuration, called *luke-reprod* has been tested. Theoretically equivalent to *luke-base*, it consists into reproducing the fine-tuning on TACRED to see if this fine-tuning is reproducible, and to have another comparison point for *luke-redetect.* Concerning *luke-detect*, several values have been tested for the size of the hidden layer, and best results have been obtained with $n = 400$. The implementation is freely accessible[3] , and the experiments were run using a Tesla V100 GPU.

Results Table 1 shows the performance for the three detailed configurations. It can be read that, contrary to our expectations, *luke-reprod* does not reproduce the results from *luke-base*, by having an F-score inferior by 1.3 points. LUKE's implementation being in Python, this is probably due to a problem in dependency versioning. However, even if the reproduction was a failure, we can observe that *luke-detect*'s F-score is superior by 0.4 points to *luke-reprod*'s one.

[3] See https://gitlab.inria.fr/hayats/luke-redect

Therefore, it can be hoped that if we were able to reproduce perfectly *luke-base*, *luke-detect* would have a better F-score.

It is interesting to point out that if *luke-base* has an overall better F-score, *luke-reprod* outperforms its precision and *luke-redetect* outperforms its recall. However, having a lower recall means having more false-negative examples, which means missing some examples expressing a relation, which we want to avoid, while having a lower precision means trying to classify a relation on examples that express none, which is also problematic. This is why we prefer F-score over precision or recall, and therefore we use *luke-base* as a relation detection module in the following experiments.

5.2 Relation Classification

We now evaluate our Concepts of Neighbours-based module individually on the Relation Classification task.

Experiment Design These experiments are made on the positive examples of TACRED, *i.e.* the examples that have an annotation other than *no_relation*. As our method does not have any use of a development corpus, we merge this corpus with the training one. We finally obtain a dataset composed of 18,446 training examples and 3,325 test examples. The quality measure usually used on TACRED is the micro-averaged F-score. However, as there is no negative class on this task, this measure does not make sense, and therefore we use accuracy.

In these experiments, as we work on a subset of TACRED we cannot compare this approach directly to other existing methods. Therefore, we compare our approach to a basic baseline in the RECENT paradigm. This baseline simply predicts, for given subject and object types, the relation type that appears the most among the training examples with the same subject and object types.

As the algorithm for the computation of Concepts of Neighbours is *anytime*, we have to choose a timeout for our experiments. In order to see how the time-out influences the classification task, several timeouts were tested between 10 and 1200 seconds. Concerning the dependency tree pruning, several values of k were tested, and the best results have been obtained with $k = 1$.Our approach was implemented in Java[4] and uses *ConceptualKNN*[5] for the computation of Concepts of Neighbours, which is based on Apache Jena[6], a Java library for the semantic web.

Results Table 2 presents the accuracy for the baseline and for our approach. First it can be observed that the baseline has an accuracy of 80.4%, which is particularly high, which means that the dataset leaves little space for progress. Then, it can be read that for any timeout, the proposed approach has a better accuracy than the baseline, surpassing it by 2.2 points for a timeout of over 300s.

[4] Accessible here: https://gitlab.inria.fr/hayats/conceptualknn-relex
[5] https://gitlab.inria.fr/hayats/jena-conceptsofneighbours
[6] https://jena.apache.org/

Table 2. Accuracy for relation classification, compared to the baseline.

Timeout (s)	10	20	30	60	120	300	600	1200
Ours	82.0	82.1	82.7	82.9	83.4	**83.6**	**83.6**	**83.6**
Baseline	80.4							

Table 3. F-score for several Relation Extraction methods on TACRED

Method	F1 score
LUKE [22]	**72.7**
BERT-LSTM-Base [20]	67.8
Ours	*66.9*
C-GCN [23]	66.4
GCN [23]	64.0

In addition, this table clearly shows a saturation phenomenon: there is an important gain when timeout gets from 10s to 120s, gain that is far smaller from 120s to 1200s. It can be intuited that this comes from the fact that most concepts are computed before 120 seconds, and only a few concepts are added after 120s. This also can be seen in the proportion of examples for which the full set of Concepts of Neighbours is computed: of less than 30% for a timeout of 10s, it rises to over 80% for a timeout of 120s and to over 99% for a timeout of over 600s. This shows that despite the anytime algorithm, most of the prediction is made on the real set of Concepts of Neighbours, and not an approximation.

5.3 Relation Extraction

Now that we have shown that our Concepts of Neighbours-based method is a valid approach for relation classification and that we have chosen a deep learning relation detection module, both can be assembled to form a full relation extraction method. In this subsection we present the experimental process to evaluate this method, as well as both quantitative and qualitative results.

Experiment Design We evaluate our two-step approach on the full TACRED dataset in order to compare it to previous approaches. To do so, according to the structure presented in Figure 3, we process the test examples of TACRED with *luke-base*, and obtain examples classified as positive or negative. Then, each example classified as positive is processed by our Concept-of-Neighbours module for relation classification.

Quantitative Results Table 3 compares our method with previous Relation Extraction methods. It shows that although our method is not competitive with pre-trained language models such as BERT or LUKE, it outperforms approaches based on graph convolution networks. Indeed, our method beats by 2.9 F-score points the basic graph convolution network (GCN) and by 0.5 points the contextualized graph convolution network (C-GCN). This is interesting because our

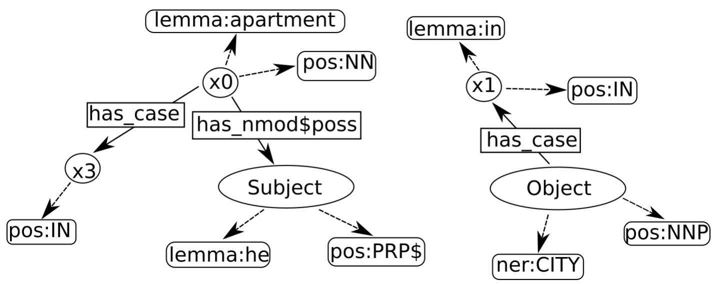

Fig. 4. Example of rule body

method and those two methods are conceptually close: both are based on the representation of sentences as a graph, both use the pruned dependency tree of the sentences, and both add to this modeling a semantic layer (a word embedding for GCN and C-GCN, WordNet for our approach). The difference between those approaches is that ours aims to provide explanations for the examples classified as positive.

Qualitative Results As mentioned above, the main advantage of our approach is its explainability. Let us take for example the sentence *"Sollecito has said he was at **his** own apartment in **Perugia**, working at his computer."* *luke-base* predicts that there is a relation between the subject (*his*) and the object (*Perugia*). As the subject is a person and the object a city, there are only three compatible relations: *per:cities_ of_ residence*, *per:city_ of_ death* or *per:city_ of_ birth*. After computation of the Concepts of Neighbours, we observe that the relation *per:city_ of_ residence* is predicted, as six rules of confidence 1 predict it, while only one rule each predicts the other two compatible relations. Figure 4 shows the body of one of those rules. It can be read as:

- The subject has lemma *he* and is the possessor of an apartment;
- The object is the name of a city in which there is something.

Even if this pattern is too specific to form a general rule, it can be infered that, knowing there is a relation between the subject and the object, we can be pretty sure that any sentence following this pattern has the relation *per:cities_ of_ residence* between its subject and its object. To complete this explanation, we can look at the training examples matching this rule. In our case, there is one sentence matching it: *"Wilbert Gibson walked from **his** apartment to the grocery store earlier this week – that's what people do in **New York City** – and thought this must be what it's like to be a celebrity."* We can see that this sentence effectively expresses the relation *per:cities_ of_ residence*, but quite implicitly. Therefore, this is interesting to see that this kind of pattern can be captured and exploited by our approach.

In practice, we observe that the rules of maximum confidence have systematically a confidence equal to 1. This is due to the fact that Concepts of Neighbours compute rules specific enough to match a few cases, and therefore to have a low extensional distance. After reviewing the explanations for ten randomly chosen correct predictions, we can observe that 56% of the 172 graph patterns seem reliable. Most of those reliable explanations are considered as such because of a lemma or a synset appearing in the graph pattern (for example the word *daughter* to characterize the relation *per:children*). In addition, we observe that the reliability of the explanations depends on the relation type. For example it can be pointed out that for an example predicting the relation *per:top_member/employee*, most of the explanations are invalid. This is caused by the fact that there is a great variety of words or formulations expressing this relation, and therefore the same one is rarely used several times. In addition, it appears that most of graph patterns are disconnected, but, as we could hope, most of the connected ones are valid.

6 Conclusion

In this article, we presented a new method for relation extraction. The core idea of this method is to combine an explainable and symbolic approach for relation classification with a deep learning method for relation detection. More precisely, we present a FCA-based approach that has shown promising results on relation classification, and we couple it with a state-of-the-art pre-trained language model fine-tuned for relation detection. Experiments have shown that this two-step approach gives promising results. In addition, this new method explains each positive prediction with interpretable rules.

In the future, work has to be made on the FCA-based relation classifier, on the modeling, by adding sequentiality for example, as well as on the concepts of neighbours, in order to use more expressive and flexible patterns. There is also work to do on the explainability, on how to display those explanations in order to make them easily readable , in order to to allow for interaction with the user.

References

1. Ayats, H., Cellier, P., Ferré, S.: Extracting Relations in Texts with Concepts of Neighbours. In: International Conference on Formal Concept Analysis (2021)
2. Ben Abacha, A., Zweigenbaum, P.: Automatic extraction of semantic relations between medical entities: a rule based approach. J. Biomedical Semantics 2 (2011)
3. Cellier, P., Charnois, T., Plantevit, M., Rigotti, C., Crémilleux, B., Gandrillon, O., Kléma, J., Manguin, J.: Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts. J. Biomedical Semantics 6, 27 (2015)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT (2019)
5. Ferré, S.: Concepts de plus proches voisins dans des graphes de connaissances. In: Ingénierie des Connaissances (IC) (2017)
6. Ferré, S.: Application of Concepts of Neighbours to Knowledge Graph Completion. Data Science 4(1), 1–28 (2021)

7. Fundel, K., Küffner, R., Zimmer, R.: RelEx - relation extraction using dependency parse trees. Bioinformatics 23(3), 365–371 (2007)
8. Giuliano, C., Lavelli, A., Romano, L.: Exploiting shallow linguistic information for relation extraction from biomedical literature. In: Conf. Eu. Chapter of the Association for Computational Linguistics. pp. 401–408 (2006)
9. Grishman, R.: Twenty-five years of information extraction. Natural Language Engineering pp. 677–692 (2019)
10. Gutierrez, C., Sequeda, J.F.: Knowledge graphs. Communications of the ACM 64(3), 96–104 (2021)
11. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: Improving Pre-training by Representing and Predicting Spans. Trans. Association for Computational Linguistics 8, 64–77 (2020)
12. Leeuwenberg, A., Buzmakov, A., Toussaint, Y., Napoli, A.: Exploring Pattern Structures of Syntactic Trees for Relation Extraction. In: Formal Concept Analysis. pp. 153–168. LNAI 9113 (2015)
13. Lyu, S., Chen, H.: Relation Classification with Entity Type Restriction (May 2021), http://arxiv.org/abs/2105.08393
14. Mallart, C., Le Nouy, M., Gravier, G., Sébillot, P.: Active Learning for Interactive Relation Extraction in a French Newspaper's Articles. In: Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications (2021)
15. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2014)
16. Martinez-Rodriguez, J.L., Hogan, A., Lopez-Arevalo, I.: Information extraction meets the Semantic Web: A survey. The Semantic Web 11(2), 255–335 (2020)
17. Meilicke, C., Chekol, M.W., Ruffinelli, D., Stuckenschmidt, H.: Anytime Bottom-Up Rule Learning for Knowledge Graph Completion. In: Int. Joint Conf. Artificial Intelligence. pp. 3137–3143 (2019)
18. Miller, G.A.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge, MA (1998)
19. Nguyen, T.H., Grishman, R.: Relation Extraction: Perspective from Convolutional Neural Networks. In: Work. Vector Space Modeling for Natural Language Processing. pp. 39–48 (2015)
20. Shi, P., Lin, J.: Simple BERT Models for Relation Extraction and Semantic Role Labeling (Apr 2019), arXiv: 1904.05255
21. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying Relations via LSTM Networks along Shortest Dependency Paths. In: Conf. Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2015)
22. Yamada, I., Asai, A., Shindo, H., Takeda, H., Matsumoto, Y.: LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In: Conf. Empirical Methods in Natural Language Processing (EMNLP) (2020)
23. Zhang, Y., Qi, P., Manning, C.D.: Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In: Conf. Empirical Methods in Natural Language Processing. pp. 2205–2215 (2018)
24. Zhang, Y., Zhong, V., Chen, D., Angeli, G., Manning, C.D.: Position-aware Attention and Supervised Data Improve Slot Filling. In: Conf. Empirical Methods in Natural Language Processing. pp. 35–45 (2017)